

## BAB II

### TINJAUAN PUSTAKA

#### A. Penelitian Terdahulu

Sistem pakar ini dikembangkan untuk meingkatkan hasil dari penelitian sebelumnya yang sudah pernah dilakukan. Adapun penelitian sebelumnya yang telah dilakukan oleh Budi Kurniawan (2011) mengenai aplikasi sistem pakar berbasis web untuk diagnose penyakit gigi dan mulut. Perbedaan dan peningkatan yang dilakukan dari penelitian sebelumnya adalah *platform* pengembangan sistem. Pada penelitian yang dilakukan oleh Budi Kurniawan *platform* yang digunakan adalah berbasis web, sedangkan pada penelitian ini *platform* yang digunakan adalah berbasis android, sehingga mudah digunakan, dan cara penanganan gejala awal yang dialami pasien dapat dilakukan dengan menggunakan obat tradisional.

Penelitian lain mengenai sistem pakar adalah sistem pakar pada perangkat *mobile* untuk mendiagnosa penyakit gigi yang telah dilakukan oleh Bambang Yowono (2010). Perbedaan dan peningkatan yang dilakukan dari penelitian sebelumnya adalah metode yang digunakan. Pada penelitian yang dilakukan oleh Bambang Yowono metode yang digunakan adalah *backward chaining*, sedangkan pada penelitian ini menggunakan metode *forward chaining*, akan tetapi *platform* yang digunakan untuk mengembangkan sistem pada penelitian sebelumnya dan penelitian ini adalah sama – sama menggunakan berbasis mobile, yang membedakan yaitu teknologi dan cara pengaksesan sistemnya. Pada penelitian sebelumnya sistem pakar ini dapat diakses melalui *Handphone* atau PDA dengan teknologi *Wireless Applicatin Protocol* (WAP), sedangkan pada penelitian ini sistem pakar dapat diakses melalui *Smartphone Android*.

Penelitian sebelumnya mengenai sistem pakar berbasis Android juga telah dilakukan oleh Quwais Alqorni Sahara (2009). Penelitian tersebut membahas tentang penyakit usus pada manusia dengan metode *certainty factor*. Jadi jika dibandingkan dengan penelitian ini maka perbedaannya adalah jenis penyakit yang akan didiagnosa dan metode yang digunakan oleh Quwais. Untuk *platform*

pengembangan sistemnya penelitian sebelumnya dan penelitian ini adalah sama – sama menggunakan *platform Smartphone* berbasis *Android*.

Dwi Indah Setiowati (2015) juga telah melakukan penelitian mengenai sistem pakar diagnosa penyakit gigi dan mulut berbasis android dengan metode probabilitas. Pada penelitian tersebut Dwi Indah menggunakan metode probabilitas untuk menyelesaikan masalah diagnosa penyakit gigi dan mulut sedangkan pada penelitian ini metode yang digunakan untuk menyelesaikan masalah adalah *forward chaining*. Untuk pengembangan sistemnya penelitian ini dan penelitian sebelumnya sama – sama menggunakan aplikasi berbasis *android*.

Penelitian mengenai sistem berbasis pengetahuan untuk kesehatan dan perawatan gigi dan mulut yang telah dilakukan oleh I Wayan Shandyasa (2011) menjelaskan bahwa selain menyediakan informasi mengenai kesehatan dan perawatan gigi dan mulut sistem tersebut juga menyediakan diagnosa penyakit yang sedang dialami oleh user. Metode penelusuran yang digunakan oleh I Wayan adalah *forward chaining*. Sedangkan *platform* yang digunakan pada sistem tersebut adalah berbasis web. Jadi perbedaan dan peningkatan jika dibandingkan dengan penelitian sebelumnya adalah jenis *platform* yang digunakan, pada penelitian ini penulis menggunakan *platform* berbasis *Smartphone* berbasis *Android*.

## **B. Aplikasi**

Menurut Dhanta (2009:32) aplikasi (*Application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan.

Sutabri (2012:147) Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

Dari pengertian di atas, dapat disimpulkan bahwa aplikasi adalah alat yang dibuat untuk melaksanakan tugas tertentu sesuai dengan fungsi dan kemampuannya.

## **C. Sistem**

Menurut Ladjamudin (2005 : 6) Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lainnya. Karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada didalam system tersebut.

### **1. Karakteristik Sistem**

Menurut Ladjamuddin (2005 : 3) suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem penghubung, masukan, keluaran, pengolahan dan saran atau tujuan. Adapun Karakteristik yang dimaksud adalah :

#### **1. Komponen Sistem :**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan.

#### **2. Batasan Sistem :**

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya.

#### **3. Lingkungan luar Sistem :**

Lingkungan luar Sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem.

#### **4. Penghubung Sistem :**

Merupakan media yang menghubungkan antara satu subsistem yang lainnya.

#### **5. Masukan Sistem :**

Masukan sistem adalah energi yang dimasukkan ke dalam sistem, masukan dapat berupa masukan perawatan dan masukan sinyal maintenance.

#### **6. Keluaran Sistem :**

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan subsistem yang lain.

7. Pengelolaan Sistem :

Suatu bagian pengolah atau sistem itu sendiri sebagai pengelolanya.

Pengelolaan yang akan berubah menjadi keluaran.

8. Sasaran Sistem :

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada.

#### **D. Sistem Pakar**

Sistem pakar merupakan cabang dari *Artificial Intelligence (AI)* yang mulai dikembangkan pada pertengahan 1960. Istilah system pakar berasal dari istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah, system pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer.

Kusrini (2006) sistem pakar yaitu keahlian dari pakar ke suatu komputer. Pengetahuan ini kemudian disimpan didalam komputer. Pada saat pengguna menjalankan komputer untuk mendapatkan informasi, sistem pakar menanyakan fakta-fakta dan membuat penalaran (*inferensi*) dan sampai pada suatu kesimpulan. Kemudian sistem pakar memberikan penjelasan (memberikan kesimpulan atas hasil konsultasi yang telah dilakukan sebelumnya).

##### **1. Manfaat Sistem Pakar**

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya adalah :

1. Meningkatkan produktivitas.
2. Membuat orang awam bekerja seperti layaknya seorang pakar.
3. Mampu menangkap pengetahuan dan kepakaran seseorang.
4. Memudahkan akses pengetahuan seorang pakar.
5. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
6. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.

7. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
8. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

## **2. Kekurangan Sistem Pakar**

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya adalah :

1. Biaya yang diperlukan untuk membuat, memelihara, dan mengembangkannya sangat mahal.
2. Sulit dikembangkan.
3. Sistem pakar tidak 100% benar.
4. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda - beda, meskipun sama - sama benar.
5. Kurangnya rasa percaya pengguna dapat menghalangi pemakaian sistem pakar.

## **3. Ciri – ciri Sistem Pakar**

Ciri – ciri dari Sistem Pakar adalah sebagai berikut :

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data – data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan – alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah / rule tertentu.
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme inferensi terpisah
7. Keluarannya bersifat anjuran
8. System dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

#### 4. Konsep Dasar Sistem Pakar

Konsep dasar dari sistem pakar yaitu meliputi keahlian (*expertise*), ahli (*experts*), pemindahan kepakaran (*transferring expertise*), inferensi (*inferencing*), aturan (*rules*) dan kemampuan memberikan penjelasan (*explanation capability*).

Keahlian (*expertise*) adalah pengetahuan yang mendalam tentang suatu masalah tertentu, dimana keahlian bisa diperoleh dari pelatihan/ pendidikan, membaca dan pengalaman dunia nyata. Ada dua macam pengetahuan yaitu pengetahuan dari sumber yang ahli dan pengetahuan dari sumber yang tidak ahli. Pengetahuan dari sumber yang ahli dapat digunakan untuk mengambil keputusan dengan cepat dan tepat.

Ahli atau Pakar (*experts*) adalah seorang yang memiliki keahlian tentang suatu hal dalam tingkatan tertentu, ahli dapat menggunakan suatu permasalahan yang ditetapkan dengan beberapa cara yang berubah-ubah dan merubahnya kedalam bentuk yang dapat dipergunakan oleh dirinya sendiri dengan cepat dan cara pemecahan yang mengesankan.

Pemindahan Kepakaran (*transferring expertise*) adalah tujuan dari sistem pakar yaitu memindahkan kepakaran dari seorang pakar ke dalam computer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu :

1. Akuisisi pengetahuan ( dari pakar atau sumber lain )
2. Representasi pengetahuan ( pada komputer )
3. Inferensi pengetahuan
4. Pemindahan pengetahuan ke pengguna

Inferensi (*inferencing*) adalah sebuah *procedure* ( program ) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

Software system pakar komersial adalah system yang berbasis *rule* ( *rule-based systems*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai *procedure* – *procedure* pemecahan masalah.

Fasilitas yang lain dari system pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan

dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari system ini memungkinkan system untuk memeriksa penalaran yang dibuat sendiri dengan menjelaskan operasi – operasinya.

Karakteristik dan kemampuan yang dimiliki oleh system pakar berbeda dengan system konvensional.

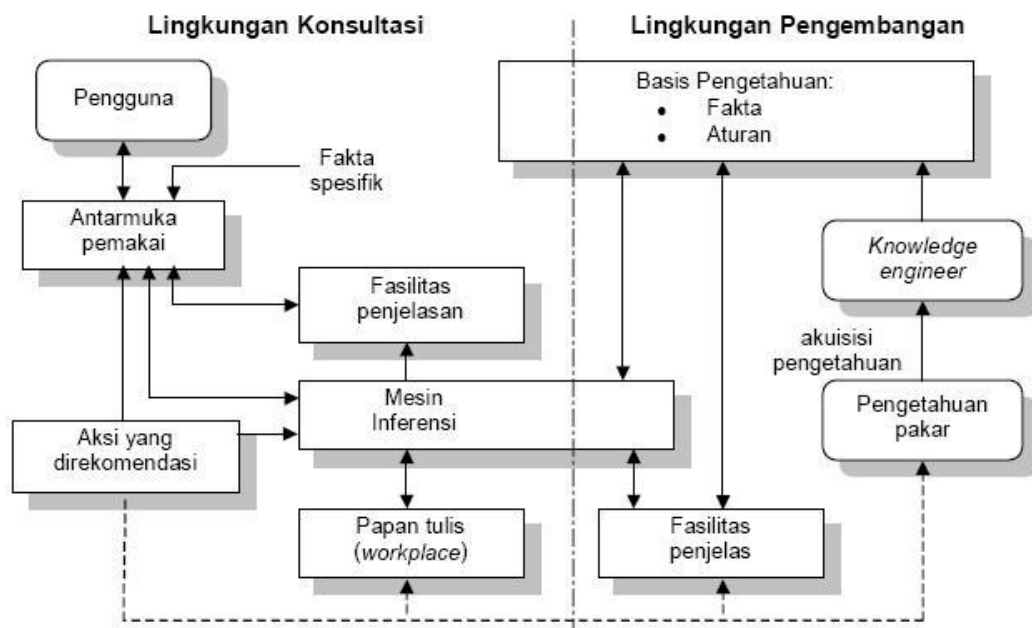
Tabel 2.1 : Perbandingan Sistem Konvensional dengan Sistem Pakar

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesannya biasanya menjadi satu dengan program	Basis pengetahuan merupakan bagian terpisah dari mekanisme inferensi
Biasanya tidak dapat menjelaskan mengapa suatu input data dibutuhkan atau bagaimana output itu diperoleh	Penjelasan adalah bagian terpenting dari sistem pakar
Pengubahan program cukup sulit dan merepotkan	Pengubahan pada aturan/kaidah dapat dilakukan dengan mudah
Eksekusi dilakukan langkah demi langkah secara algoritmik	Eksekusi dilakukan pada keseluruhan basis pengetahuan secara heuristic atau logic
Menggunakan data	Menggunakan pengetahuan
Program tidak pernah salah (kecuali pemrogramnya yang salah)	Program bisa saja melakukan kesalahan

Sumber : Buku Konsep Kecerdasan Buatan tahun 2006

## 5. Struktur Sistem Pakar

Sistem pakar terdiri dari dua bagian pokok, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seorang yang bukan ahli untuk berkonsultasi.



Sumber : <http://elib.unikom.c.id>

Gambar 2.1 : Struktur Sistem Pakar

Keterangan :

### 1. Akuisisi Pengetahuan

Akuisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh computer.

### 2. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu :



- a. Fakta, misalnya situasi atau permasalahan yang ada
  - b. *Rule* ( Aturan ), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah
3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan.
4. Daerah Kerja (*Workplace*)

*Workplace* merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil antara kesimpulan yang dicapai.

Terdapat 3 tipe keputusan yang dapat direkam pada *workplace* yaitu :

  - a. Rencana : bagaimana menghadapi masalah
  - b. Agenda : aksi-aksi potensial yang sedang menunggu untuk dieksekusi
  - c. Solusi : calon aksi yang akan dibangkitkan
5. Antarmuka Pengguna (*User Interface*)

User interface merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem dan sebaliknya.
6. Fasilitas Penjelasan

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai.
7. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

## 8. Pengguna ( *User* )

Pada umumnya pengguna system pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.

## 6. **Rule Sebagai Teknik Representasi Pengetahuan**

Setiap rule terdiri dari dua bagian, yaitu bagian IF disebut sebagai *evidence* (Fakta-fakta) dan bagian THEN disebut *Hipotesis* atau Kesimpulan.

Syntax *Rule* adalah : **IF E THEN H**

Secara umum *rule* yang mempunyai *evidence* lebih dari satu yang dihubungkan oleh kata penghubung **AND** atau **OR** atau kombinasi keduanya. Tetapi sebaiknya biasakan menghindari penggunaan **AND** dan **OR** secara sekaligus dalam satu *rule*.

**IF ( E<sub>1</sub> AND E<sub>2</sub> AND E<sub>3</sub> ..... AND E<sub>n</sub> ) THEN H**

**IF ( E<sub>1</sub> OR E<sub>2</sub> AND E<sub>3</sub> ..... OR E<sub>n</sub> ) THEN H**

Selain itu satu *evidence* bisa juga mempunyai *hipotesis* lebih dari satu. Seperti di bawah ini :

**IF E THEN (H<sub>1</sub> AND H<sub>2</sub> AND H<sub>3</sub> ..... AND H<sub>n</sub>)**

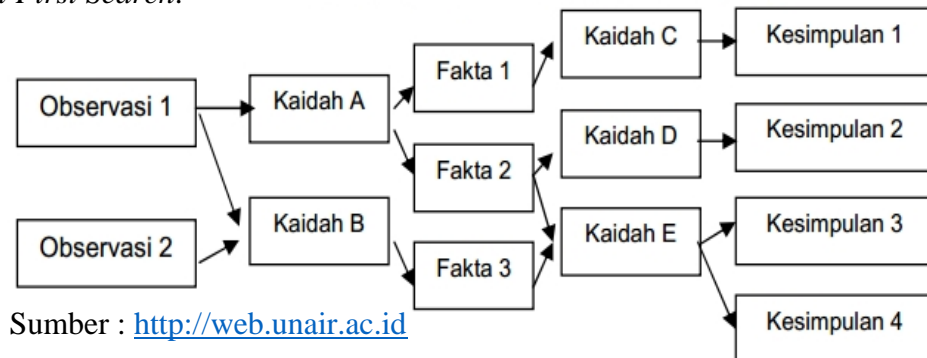
## 7. **Teknik Inferensi *Forward Chaining* dan *Backward Chaining***

Pada sistem pakar berbasis *rule*, domain pengetahuan direpresentasikan dalam sebuah kumpulan rule berbentuk IF – THEN, sedangkan data direpresentasikan dalam sebuah kumpulan fakta – fakta tentang kejadian saat ini. Mesin Inferensi membandingkan masing – masing rule yang tersimpan dalam basis pengetahuan dengan fakta – fakta yang terdapat dalam database. Jika bagian IF (kondisi) dari rule cocok dengan fakta, maka rule dieksekusi dan bagian THEN (aksi) diletakkan dalam database sebagai fakta baru yang ditambahkan.

### **a. Pelacakan Kedepan (*Forward Chaining*)**

*Forward Chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta – fakta tersebut dengan bagian IF dari rule IF – THEN. Bila ada fakta yang cocok dengan bagian IF, maka *rule*

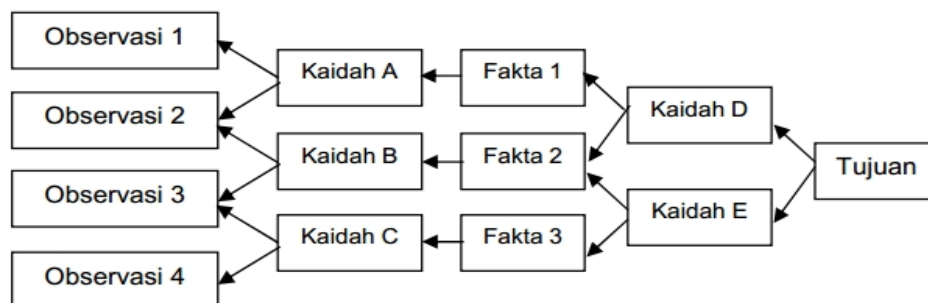
tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang dieksekusi. Metode pencarian yang digunakan adalah *Depth-First Search* (DFS), *Breadth-First Search* (BFS), atau *Best First Search*.



Gambar 2.2 : Diagram Pelacakan Kedepan

#### b. Pelacakan Kebelakang (*Backward Chaining*)

*Backward Chaining* adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari goal (yang berada dibagian THEN dari rule IF-THEN), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta – fakta yang ada cocok dengan premis – premis dibagian IF. Jika cocok, rule dieksekusi, kemudian hipotesis di bagian THEN ditempatkan di basisdata sebagai fakta baru. Jika tidak cocok, simpan premis dibagian IF ke dalam stack sebagai sub goal. Proses berakhir jika goal ditemukan atau tidak ada rule yang bisa membuktikan kebenaran dari sub goal atau goal.



Gambar 2.3 : Diagram Pelacakan Kebelakang

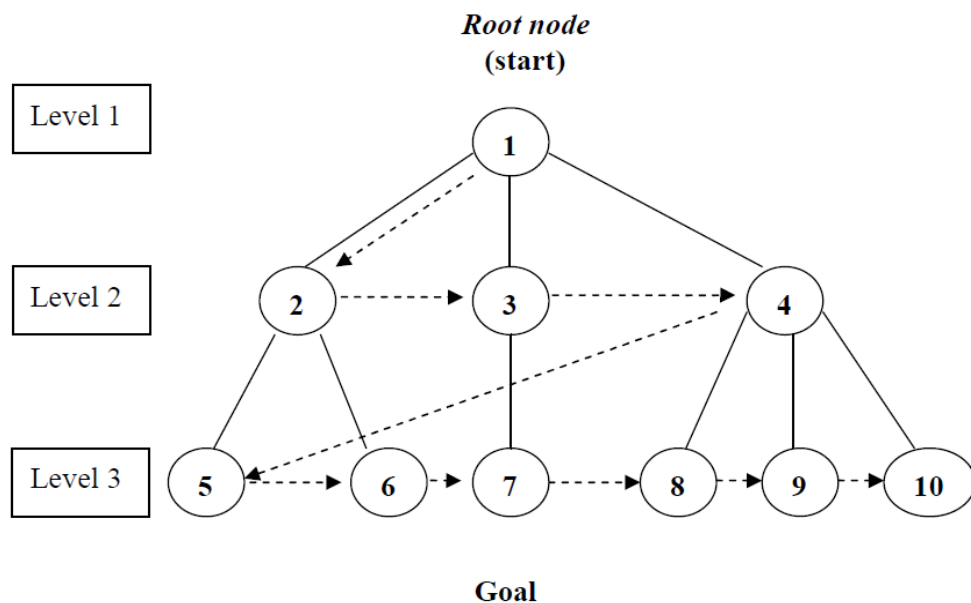
## 8. Metode Pencarian

Pada dasarnya teknik pencarian dapat dibagi menjadi 2 (dua) kelompok besar, yaitu pencarian buta (blind search) dan pencarian terbimbing (*heuristic search*). Untuk mengukur performa metode pencarian, terdapat 4 kriteria yang dapat dilakukan yaitu :

1. *Completeness*
2. *Time Complexity*
3. *Space Complexity*
4. *Optimality*

### a. *Breadth – First Search (BFS)*

Metode penelusuran ini memeriksa semua node (simpul) pohon pencarian, dimulai dari simpul akar. Simpul-simpul dalam tingkat diperiksa seluruhnya sebelum pindah ke simpul di tingkat selanjutnya. Proses ini bekerja dari kiri ke kanan, baru bergerak ke bawah. Ini berlanjut sampai ke titik tujuan (*goal*).

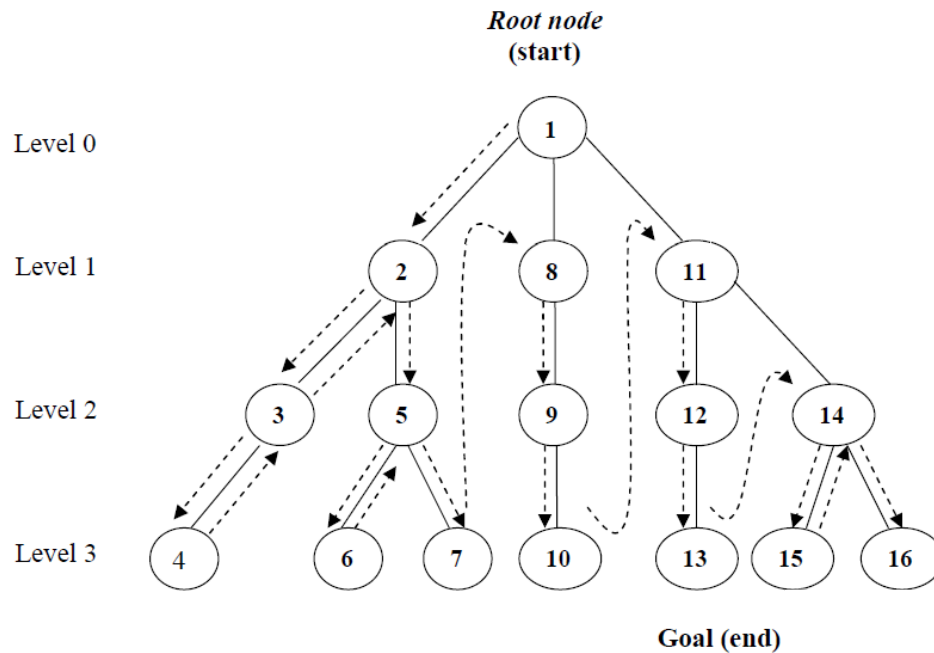


Sumber : <http://elib.unikom.ac.id>

Gambar 2.4 : *Breadth – First Search*

### b. *Depth – First Search (DFS)*

Metode ini memulai penelusuran dari node sampai simpul akar, selanjutnya menuju ke bawah dulu baru bergerak ke samping dari kiri ke kanan, proses ini akan berlanjut sampai ditemukan simpul tujuan (*goal*).

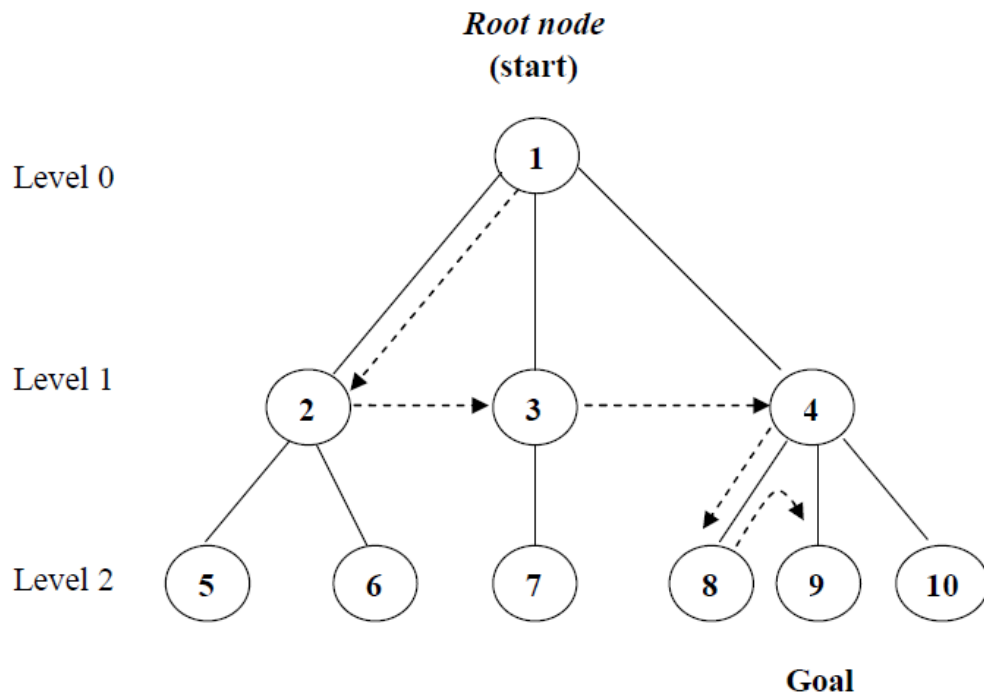


Sumber : <http://elib.unikom.ac.id>

Gambar 2.5 : *Depth – First Search*

### c. *Best First Search*

Metode best first search merupakan kombinasi dari depth first search dan breadth first search yang mana pencarian diperbolehkan mengunjungi node yang ada di level yang lebih rendah asalkan node ini memiliki nilai heuristic yang lebih baik.



Gambar 2.6 : *Best – First Search*

### E. Diagnosis

Menurut Webster (2007) Diagnosis adalah proses menentukan hakekat dari pada kelalaian atau ketidakmampuan dengan ujian dan melalui ujian tersebut dilakukan suatu penelitian yang hati-hati terhadap fakta-fakta untuk menentukan masalahnya.

Menurut Hariman (2008) Diagnosis adalah suatu analisis terhadap kelainan kelainan atau salah penyesuaian dari simptom-simptomnya.

Dari penjelasan diatas dapat disimpulkan bahwa diagnosis adalah suatu cara menganalisis suatu kelainan dengan mengamati gejala-gejala yang nampak dan dari gejala tersebut dicari factor penyebab kelainan tadi.

### F. Penyakit

Menurut Ibrahim (2008) Penyakit adalah suatu keadaan abnormal dari tubuh atau pikiran yang menyebabkan ketidaknyamanan, disfungsi atau kesukaran

terhadap orang dipengaruhinya, orang-orang biasa berkonsultasi dengan seorang dokter.

Menurut Lukhas (2009) Penyakit adalah keadaan tidak normal pada badan atau minda yang menyebabkan ketidak selarasan, disfungsi, atau tekanan/stress kepada orang yang terbabit atau berhubungan rapat dengannya.

## **G. Gigi dan Mulut**

Menurut Pratiwi (2016) Gigi adalah bagian terkeras dari tubuh manusia yang komposisinya bahan organik dan airnya sedikit sekali, sebagian besar terdiri dari bahan anorganik sehingga tidak mudah rusak, terletak dalam rongga mulut yang terlindung dan basah oleh air liur.

Gigi adalah tulang keras dan kecil-kecil berwarna putih yang tumbuh tersusun, berakar didalam gusi dan berfungsi untuk mengunyah dan menggigit. Gigi adalah jaringan tubuh yang paling keras dibanding yang lainnya. Strukturnya berlapis-lapis mulai dari email yang amat keras, dentin (tulang gigi) di dalamnya, pulpa yang berisi pembuluh darah, pembuluh saraf, dan bagian lain yang memperkokoh gigi. Namun demikian, gigi merupakan jaringan tubuh yang mudah sekali mengalami kerusakan. Ini terjadi ketika gigi tidak memperoleh perawatan semestinya. Proses kerusakan gigi geligi diawali dengan adanya lubang gigi atau disebut juga karies.

Mulut merupakan jalan masuk untuk sistem pencernaan. Bagian dalam dari mulut dilapisi oleh selaput lendir. Pengecapan dirasakan oleh organ perasa yang terdapat di permukaan lidah. Pengecapan relatif sederhana, terdiri dari manis, asam, asin dan pahit.

### **1. Karies Gigi**

Karies adalah penyakit pada jaringan keras gigi yang disebabkan oleh kerja mikroorganisme pada karbohidrat yang dapat diragikan. Karies ditandai dengan adanya demineralisasi mineral-mineral email dan dentin, diikuti kerusakan bahan-bahan organiknya. Ketika makin mendekati pulpa, karies menimbulkan perubahan dalam bentuk dentin reaksioner dan pulpitis (mungkin disertai nyeri) dan bisa

berakibat terjadinya invasi bakteri dan kematian pulpa. Jaringan pulpa mati yang terinfeksi ini selanjutnya akan menyebabkan perubahan di jaringan periapikal.

#### **a. Faktor yang dapat membantu terjadinya karies gigi**

Mudah tidaknya seseorang terserang karies gigi tidak hanya disebabkan oleh satu faktor saja tetapi ditentukan oleh banyak faktor. Orang yang bertempat tinggal pada satu daerah, belum tentu mempunyai jumlah karies yang sama. Beberapa faktor yang dapat membantu terjadinya karies gigi :

1. Gigi :
  - a) Campuran bahan-bahan pembentuk gigi
  - b) Bentuk morfologi gigi
  - c) Posisi gigi-gigi dalam deretan.
2. Saliva :
  - a) Campuran bahan-bahan yang terkandung di dalamnya
  - b) derajat keasaman
  - c) Jumlah/ volume
  - d) faktor anti bakteri
3. Makanan :
  - a) Macam/ jumlah
  - b) Kandungan karbohidrat
  - c) Kandungan vitamin

#### **b. Kedalaman Karies**

1. Karies Superfisialis

Diagnosa : IRITASI PULPA

- a. Pengertian

Suatu keadaan di mana lapisan email telah mengalami kerusakan sampai batas Dentino Enamel Junction yang merupakan tempat terakhir dari ujung-ujung syaraf yang sudah dapat dirangsang.

- b. Penyebab-penyebab

1. Plak
2. Faktor mekanis, misal; cara menyikat gigi yang salah



- c. Gejala-gejala
    - 1. Linu bila terkena rangsangan dingin, manis, asam dan bila terkena sikat gigi .
    - 2. Rasa linu hilang bila rangsangan dihilangkan
  - d. Rencana Perawatan
 

Tumpatan, sesuai indikasi
2. Karies Media
- Diagnosa : HIPEREMI PULPA
- a. Pengertian
 

Suatu keadaan di mana kerusakan sudah sampai ke lapisan dentin, merupakan keadaan lanjut dari iritasi pulpa
  - b. Penyebab - penyebab
    - 1. Plak
    - 2. Trauma
  - c. Gejala-gejala
    - 1. Terasa linu bila kena rangsang manis, asam, dingin, panas (kadang-kadang)
    - 2. Bila rangsang dihilangkan, rasa linu tetap bertahan  $\frac{1}{2}$  - 1 menit
    - 3. Kadang-kadang linu bila kemasukan makanan
  - d. Proses terjadinya (secara Histopatologis)
 

Akibat masuknya toksin ke dalam kamar pulpa melalui saluran dentin, maka pulpa memberikan reaksi berupa pelebaran pembuluh darah dlm pulpa, sehingga siklus darah pd pulpa bertambah. Pada kasus hiperemi pulpa sering terjadi dentin tertier (sklerotik).
  - e. Rencana Perawatan
 

Tumpatan sesuai indikasi (pada kartu status ditulis pro-konservasi)

### **c. Pencegahan Karies**

Mengingat penyakit ini memerlukan bakteri plak, substrat karbohidrat, dan permukaan gigi yang rentan, maka terdapat tiga cara dalam pencegahan karies, yaitu :

1. Hilangkan substrat karbohidrat
2. Meningkatkan ketahanan pejamu
3. Hilangkan bakteri plak.

## **2. Gingivitis**

Gingivitis adalah peradangan pada gingiva yang menunjukkan adanya tanda-tanda penyakit / kelainan pada gingiva.

Gingivitis disebabkan oleh plak dan dipercepat adanya faktor iritasi lokal dan sistemik. Macam – macam iritasi lokal yang dapat menyebabkan gingivitis :

- a) *materia alba*
- b) karang gigi
- c) *overhanging filling* (tambalan berlebih)
- d) obat, misal : Arsen, Phenol

### **a. Gingivitis Marginalis**

Gingivitis marginalis adalah peradangan gingiva bagian marginal yang merupakan stadium yang paling awal dari penyakit periodontal.

Macam-macam gingivitis berdasarkan lamanya penyakit :

- a) Gingivitis marginalis akut

Tanda-tanda klinis :

1. warna merah
2. mengkilat
3. bengkak
4. mudah berdarah
5. terdapat eksudat
6. sakit

b) Gingivitis marginalis kronis

Tanda-tanda klinis :

1. warna merah bisa berubah menjadi kebiruan
2. bengkak
3. tidak mudah berdarah
4. biasanya tidak sakit
5. terdapat eksudat

Faktor-faktor predisposisi gingivitis :

- a) Hormonal
- b) Defisiensi vitamin : *Scorbutic gingivitis*
- c) Penyakit : *ANUG (Acute Necrotizing Ulcerative Gingivitis)*

Rencana perawatan Gingivitis :

1. Pencegahan :
  - a) *Oral hygiene* diperbaiki (plak kontrol)
  - b) gizi ditingkatkan
2. Pengobatan :
  - a) pemberian obat kumur yang bersifat antiseptik dan analgetik
  - b) pembersihan karang gigi (*scaling*)

## **H. DFD (*Data Flow Diagram*)**

Menurut Ladjamudin (2005:64) dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi *Data Flow Diagram* adalah diagram aliran data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil.

Menurut Jogiyanto (2005) dalam bukunya Basia Data ada beberapa simbol digunakan pada DFD untuk mewakili :

1. Kesatuan Luar (*External Entity*)

Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem

lain yang berada pada lingkungan luarnya yang memberikan *input* atau menerima *output* dari sistem.

2. Arus Data (*Data Flow*)

Arus Data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir di antara proses, simpan data dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

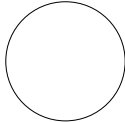



3. Proses (*Process*)

Proses (*process*) menunjukkan pada bagian yang mengubah input menjadi output, yaitu menunjukkan bagaimana satu atau lebih input diubah menjadi beberapa output. Setiap proses mempunyai nama, nama dari proses ini menunjukkan apa yang dikerjakan proses.

4. Simpanan Data (*Data Store*)

*Data Store* merupakan simpanan dari data yang dapat berupa suatu file atau database pada sistem komputer.

Tabel 2.2 : Simbol-simbol DFD

No	Symbol	Nama	Keterangan
1		Proses	Menunjukkan kegiatan/kerja yang dilakukan oleh orang, mesin atau komputer.
2		Simbol Data Flow (arus data)	Menunjukkan arus dari proses.
3		Eksternal Entity	Menunjukkan ntitas/entity
4		Data Store	Simpanan data

Sumber : Buku Analisis Desain Sistem Informasi tahun 2005


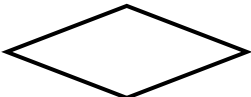


## I. ERD (*Entity Relationship Diagram*)

Menurut Ladjamudin (2005:84) *Entity Relationship Diagram* (ERD) adalah diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam system bisnis.

Adapun symbol – symbol dari *Entity Relationship Diagram* adalah sebagai berikut :

1. Persegi Panjang : Menyatakan himpunan / entitas
2. Lingkaran / elips : Menyatakan atribut (atribut yang berfungsi key digaris bawah).
3. Belah Ketupat : Menyatakan himpunan relasi
4. Garis : Sebagai Penghubung antar himpunan relasi dengan Himpunan entitas dan himpunan entitas dengan atributnya.

Tabel 2.3 : Simbol-simbol *Entity Relationship Diagram*

No	Symbol	Nama	Keterangan
1		<i>Entity / Entitas</i>	Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan
2		<i>Relationship</i>	Hubungan yang dapat terjadi antara satu entitis atau lebih
3		Atribut	Karakteristik dari entitas atau Relationship yang menyediakan penjelasan detail entitas atau relation
4		<i>Link</i>	Baris sebagai penghubung antara himpunan, relasi dan himpunan entitas dan atributnya

Sumber : Buku Analisis Desain Sistem Informasi tahun 2005

## J. *Flowchart*

Menurut Ladjamudin (2006:265) dalam bukunya *Rekayasa Perangkat Lunak* Flowchart adalah bagan – bagan yang mempunyai arus yang menggambarkan langkah – langkah penyelesaian suatu masalah. Flowchart cara penyajian suatu algoritma. Ada dua macam flowchart yang menggambarkan proses dengan computer, yaitu :

### 1. System Flowchart

Bagan yang memperlihatkan urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.

### 2. Program Flowchart

Bagan yang memperlihatkan urutan instruksi yang digambarkan dengan symbol tertentu untuk memecahkan masalah dalam suatu program.

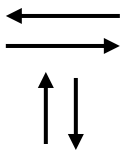
#### a. Simbol – simbol *Flowchart*



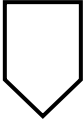
Flowchart disusun dengan symbol. Symbol ini dipakai sebagai alat bantu menggambarkan proses di dalam halaman. Symbol – symbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok yaitu :

#### 1. *Flow Direction Symbols* (Simbol Penghubung/alur)

Simbol yang digunakan untuk menghubungkan antara symbol yang satu dengan yang lainnya. Simbol ini juga disebut *connecting line*, Simbol tersebut adalah :

Tabel 2.4 : *Flow Direction Symbols*

No	Symbol	Nama	Keterangan
1		Arus / <i>Flow</i>	Untuk menyatakan jalannya arus suatu proses

2		<i>Communication link</i>	Untuk menyatakan bahwa adanya transisi suatu data atau informasi dari suatu lokasi ke lokasi lainnya
3		<i>Connector</i>	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembaran sama
4		<i>Offline Connector</i>	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman atau lembaran yang berbeda


---


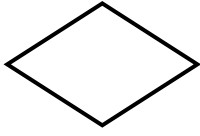
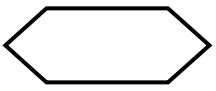
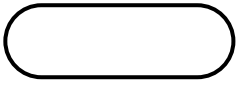
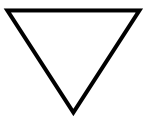

Sumber : Buku Analisis Desain Sistem Informasi tahun 2005

## 2. *Processing Symbols* (Simbol Proses)

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur. Symbol – symbol tersebut adalah :

Tabel 2.5 : *Processing Symbols*

No	Symbol	Nama	Keterangan
1		Proses	Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer biasanya menghasilkan perubahan terhadap data atau informasi

2		Symbol manual	Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh computer (manual)
3		Decision / Logika	Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan 2 kemungkinan jawaban, YA / TIDAK
4		Predefined Process	Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
5		Terminal	Untuk menyatakan permulaan atau akhir suatu program
6		Offline Storage	Untuk menunjukkan bahwa data dalam symbol ini akan disimpan ke suatu media tertentu
7		Manual Input	Untuk memasukkan data secara manual dengan menggunakan online keyword

---

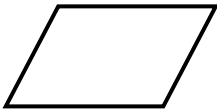


Sumber : Buku Analisis Desain Sistem Informasi tahun 2005



### 3. *Input / Output Symbols* (Simbol Input – output)

Symbol yang menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*. Symbol – symbol tersebut adalah :

Tabel 2.6 : *Input – Output Symbols*

No	Symbol	Nama	Keterangan
1		<i>Input / output</i>	Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
2		<i>Disk Storage</i>	Untuk menyatakan input berasal dari disk atau optput disimpan ke disk
3		Document	Untuk menyetak dokumen

Sumber : Buku Analisis Desain Sistem Informasi tahun 2005

### K. *Unified Modeling Language* (UML)

Menurut Shalahuddin (2013:133) *UML* (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *objectoriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.

Menurut Widodo (2011:6-7) menjelaskan tentang kegunaan UML adalah sebagai berikut :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Tabel 2.7 : Tipe Diagram UML

No	Diagram	Tujuan
1	<i>Class</i>	Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi
2	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen
3	<i>Use case</i>	Diagram ini memperlihatkan himpunan <i>use case</i> dan aktor-aktor (suatu jenis khusus dari kelas)
4	<i>Sequence</i>	Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu
5	<i>Communication</i>	Sebagai pengganti diagram kolaborasi <i>UML</i> 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan
6	<i>Statechart</i>	Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status ( <i>state</i> ), transisi, kejadian serta aktivitas
7	<i>Activity</i>	Tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem

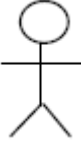
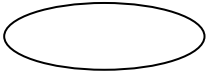

8	<i>Component</i>	Memperlihatkan organisasi serta kebergantungan sistem / perangkat lunak pada komponen-komponen yang telah ada sebelumnya
9	<i>Deployment</i>	Memperlihatkan konfigurasi saat aplikasi dijalankan ( <i>run-time</i> )

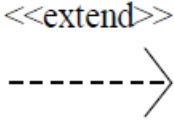

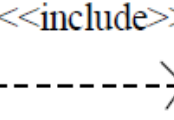
Sumber: Widodo dan Herlawati (2011:10-12)

## 1. *Use case Diagram*

Menurut Shalahuddin (2013:155) *use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut :

Tabel 2.8 : Simbol *Use case Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

4		<i>Extend</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Hubungan generalisasi dan Spesialisasi (umum-khusus) antara</p>
5		<i>Generalization</i>	<p>dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
6		<i>Include</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>

Sumber : Rosa dan Shalahuddin (2013:156-158)

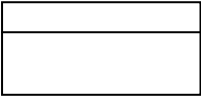




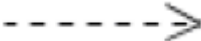

Komponen pembentuk diagram *use case* adalah:

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use case*, aktivitas / sarana yang disiapkan oleh bisnis / sistem.
3. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

## 2. *Class Diagram*

Shalahuddin (2013:141) menjelaskan tentang *class diagram* yaitu menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut :

Tabel 2.9 : Simbol *Class* Diagram







No	Simbol	Nama	Deskripsi
1		<i>Class</i>	Kelas pada struktur sistem
2		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

Sumber: Rosa dan Shalahuddin (2013:146-147)

### 3. Activity Diagram

Menurut Shalahuddin (2013:161) menjelaskan tentang *activity* diagram yaitu menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Adapun simbol-simbol yang digunakan dalam *activity* diagram adalah sebagai berikut :

Tabel 2.10 : Simbol *Activity* Diagram



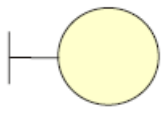



No	Simbol	Nama	Deskripsi
1		Status awal	Status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
3		Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir	Status akhir yang dilakukan sebuah sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Rosa dan Shalahuddin (2013:162-163)

#### 4. *Sequence* Diagram

Menurut Shalahuddin (2013:165) Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. *Sequence* diagram menunjukkan urutan *event* kejadian dalam suatu waktu. Komponen *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. Adapun simbol - simbol yang digunakan dalam *sequence* diagram adalah sebagai berikut :

Tabel 2.11 : Simbol *Sequence Diagram*

No	Simbol	Nama	Deskripsi
1		<i>An Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
2		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah menggambarkan dari <i>form</i>
4		<i>Control Class</i>	Menggambarkan enghubung antara boundary dengan tabel
5		<i>A focus of control</i>	Menggambarkan tempat mulai Dan berakhirnya sebuah <i>message</i> (pesan)
6		<i>A line of life</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber: Rosa dan Shalahuddin (2013:162-163)

## L. Android

Menurut Safaat (2011 : 1) dalam bukunya Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka.

Safaat (2011 : 3) menyatakan android merupakan *platform* yang lengkap, terbuka dan bebas yang artinya :

- a. Lengkap artinya para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* android. Sistem operasinya aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan peluang untuk pengembangan aplikasi
- b. Terbuka artinya platform android disediakan melalui lisensi terbuka (*open source*) sehingga pengembang dapat dengan bebas mengembangkan aplikasi.
- c. Bebas artinya tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Aplikasi android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.
- d. Aplikasi android sendiri dikembangkan pada sistem operasi berikut :
  1. Windows XP.
  2. Vista/Seven.
  3. Mac OS X (Mac OS X 10.4.8 atau lebih baru).
  4. Linux.

**a. Android SDK ( *Software Development Kit* )**

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android, Android SDK menggunakan bahasa pemrograman Java, untuk lebih mengembangkan aplikasi Android, Android SDK disediakan untuk sistem operasi Mac Os X10.4.8 atau lebih, Windows XP, Vista, 7 dan 8, serta Linux yang telah direlease oleh pihak Google.

**b. Fitur – fitur Android**

Menurut Supriyanto ( 2012 : 10 ) fitur-fitur android sebagai berikut :

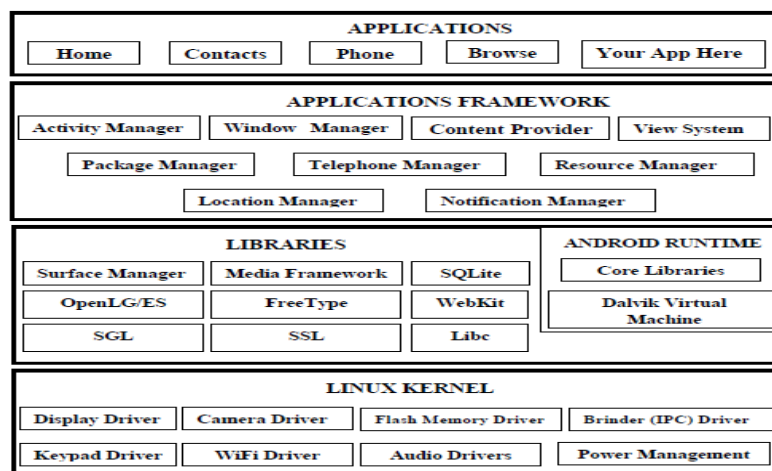
1. Penyimpanan (*storage*) menggunakan SQLite yang merupakan database relational yang ringan untuk menyimpan data.



2. Koneksi (*connectivity*) mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (termasuk A2DP dan AVRCP), WiFi, LTE dan WiMAX.
3. Pesan (*Messaging*) mendukung SMS dan MMS.
4. Web *browser* menggunakan *open source* WebKit termasuk di dalamnya *engine* Chrome V8 Java Script.
5. Media yang didukung antara lain: H.263, H.264 (3GP atau MP4), MPEG – 4 SP, AMR, AMR-WB (3GP container), ACC, HE-ACC (MP4 atau 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF dan BMP.
6. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
7. *Hardware* terdapat *accelerometer* sensor, camera, digital kompas, *proximity* sensor dan GPS.
8. *Multi-touch*
9. *Multi-tasking*
10. Dukungan *flash*

### c. Arsitektur Android

Menurut Supriyanto (2012 : 11) arsitektur android dapat dijelaskan dan digambarkan sebagai berikut :



Sumber : [www.elib.unikom.ac.id](http://www.elib.unikom.ac.id)

Gambar 2.7 : Arsitektur Android

Penjelasan mengenai gambar di atas sistem operasi android terbagi menjadi :

- a. *Applications* : layer dimana akan berhubungan dengan aplikasi saja. Biasanya aplikasi yang didownload dan diinstal dan semua aplikasi yang dibuat terletak pada tingkat *applications* seperti ditemui : *phone, contact, browser*.
- b. *Applications framework* : semacam built-in yang tertanam dalam sistem operasi android sehingga pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.
- c. *Libraries* : semua kode program yang menyediakan layanan-layanan utama sistem operasi android. Contoh *library* SQLite yang menyediakan dukungan database sehingga aplikasi android dapat untuk menyimpan data.
- d. *Android Runtime* : kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi android. Layer yang membuat aplikasi android dapat dijalankan prosesnya menggunakan implementasi *linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. *Runtime* android terbagi menjadi dua yaitu :
  1. *Core Libraries* : aplikasi android dibangun dalam bahasa java, sementara dalvik sebagai virtual mesinnya bukan virtual mesin java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java yang ditangani oleh core libraries.
  2. *Dalvik Virtual Machine* : virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangannya yang mampu membuat *linux kernel* untuk melakukan threading dan manajemen tingkat rendah.
- e. *Linux Kernel* : layer inti operating sistem android itu berada. Berisi file-file sistem yang mengatur *system processing, memory, resource, drivers* dan sistem-sistem android lainnya. *Linux kernel* yang dipakai android itu *linux kernel* release 2.6.

## M. Android Studio

Android Studio adalah sebuah lingkungan pengembangan terpadu (IDE) untuk mengembangkan untuk platform Android. Hal itu disampaikan pada tanggal 16 Mei, 2013 pada Google I / O conference oleh Product Manager Google, Katherine Chou. Android Studio tersedia secara bebas di bawah Lisensi Apache 2.0.

Android Studio adalah pada awal tahap preview akses mulai dari versi 0.1 Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada bulan Juni 2014. Yang pertama membangun stabil dirilis pada bulan Desember 2014, mulai dari versi 1.0.

Berdasarkan JetBrains 'IntelliJ IDEA software, Android Studio dirancang khusus untuk pengembangan Android. Ini tersedia untuk di-download pada Windows, Mac OS X dan Linux, dan diganti Eclipse Pengembangan Android Tools (ADT) sebagai IDE utama Google untuk pengembangan aplikasi Android asli.

## N. *SQLite*

*SQLite* adalah fasilitas yang digunakan untuk membuat *database* yang disediakan oleh Android yang secara *default* sudah tersedia di dalam *library* Android. Untuk keperluan operasi *database* pada *smartphone* atau *tablet* Android, *SQLite* sangat memadai karena ukurannya yang kecil, cepat dan ringan dalam hal sumber daya.

Menurut Djunaidi (2011) *SQLite* adalah salah satu *software* yang *embedded*, kombinasi *SQL interface* dan penggunaan *memory* yang sangat sedikit dengan kecepatan yang sangat cepat merupakan salah satu keunggulan *SQLite*. *SQLite* di Android termasuk dalam *Android Runtime*, sehingga setiap versi dari Android dapat membuat *database* dengan *SQLite*.

Sintak SQL terbagi menjadi 2 kategori yaitu:

### 1. *Data Definition Language* (DDL)

*Data Definition Language* (DDL) digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, atau *user*. Secara umum *Data*

*Definition Language* (DDL) yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yang sudah ada, dan DROP untuk menghapus objek. *Data Definition Language* (DDL) biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

2. *Data Manipulation Language* (DML)

*Data Manipulation Language* (DML) merupakan kumpulan perintah SQL yang digunakan untuk proses pengolahan isi data di dalam tabel seperti memasukkan, merubah dan menghapus isi data - dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek *database*.

Perintah-perintah *Data Manipulation Language* (DML) yang umum digunakan adalah SELECT untuk menampilkan data, INSERT untuk menambahkan data baru, UPDATE untuk mengubah data yang sudah ada, DELETE untuk menghapus data.

*SQLite* juga dapat diintegrasikan dengan bahasa – bahasa pemrograman lainnya, yaitu :

1. *SQLite* termasuk dalam *framework REALbasic*, yang memungkinkan aplikasi yang dikembangkan dengan menggunakan *REALbasic* dapat memanfaatkan basisdata *SQLite*
2. Pustaka *SQLite* bisa digunakan secara langsung pada bahasa C/C++, namun untuk Tcl dan beberapa bahasa pemrograman berbasis *script* juga tersedia.
3. Modul DBI/DBD untuk Perl juga tersedia pada CPAN DBD *SQLite*, namun modul ini bukanlah antarmuka dengan *SQLite* melainkan memasukkan secara keseluruhan dalam modul tersebut.
4. Modul Phyton juga tersedia (PySQLite) yang diimplementasikan pada DB API Phyton versi 2.0 (PEP 249).
5. PHP dimulai dengan PHP5 telah memasukkan *SQLite*, versi PHP4 sebelumnya bisa juga digunakan untuk mengakses *SQLite*, namun modul *SQLite* tidak dimasukkan secara standar bawaan.
6. Lazarus dari versi 0.9.8 dan Free Pascal 2.0.0.

7. Meskipun Borland tidak mempacketkan *SQLite* secara standar bawaan, Delphi sudah mendukung *SQLite* juga menggunakan pustaka yang dibuat oleh pihak ketiga (Auducom dan Zeos).
8. *SQLite* juga termasuk dalam paket yang *bundle* secara standar bawaan pada Mac OS X, dan digunakan sebagai salah satu pilihan mekanisme penyimpanan data pada API Apple.

Beberapa fitur – fitur dalam *SQLite* yaitu :

1. Memiliki transaksi yang bersifat *atomic*, konsistensi basisdata, isolasi, dan durabilitas (dalam Bahasa Inggris lebih sering disebut ACID).
2. Tanpa konfigurasi, tanpa instalasi dan administrasi.
3. Mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada SQL-92.
4. Mendukung *database* hingga berukuran terabyte *database* dan berukuran gigabyte string.
5. *SQLite* memiliki jejak kode kecil, membuat efisiensi penggunaan memori, ruang *disk*.
6. Mudah untuk penggunaan API.
7. Mendukung banyak sistem operasi : Unix (Linux dan Mac OS X), OS / 2, dan Windows (Win32 dan WinCE)
8. *SQLite* pilihan populer untuk mesin *database* di ponsel, PDA, MP3 *player*, *set-top box*, dan *gadget* elektronik lainnya.